07.21-00

A

Docket No.: MONG-00-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Tu, et al.

Serial No. [Not yet assigned]

Filed: July 19, 2000

For: METHOD AND APPARATUS FOR A SECURE REMOTE ACCESS SYSTEM

) Art Unit:
)
) Examiner:

CERTIFICATE OF MAILING
"Express Mail" mailing label no: EK175816603US
Date of Deposit: July 19, 2000
I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee", service under 37 CFR 1.10 on the date indicated above and is addressed to:
Assistant Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

Date: 7-19-00 Teri Muir
Teri Muir

TRANSMITTAL LETTER

Honorable Assistant Commissioner
for Patents
Box Patent Application
Washington, D.C. 20231

Sir:

Enclosed for filing please find the patent application for an invention entitled, "METHOD AND APPARATUS FOR A SECURE REMOTE ACCESS SYSTEM", filed on behalf of Monggo, Inc., assignee from inventors Edgar Allan Tu and Eric Pang, including 40 pages of specification, 3 pages of claims, 10 sheets of drawing figures, and 1 page of Abstract.

The attorney's Docket Number is MONG-00-002.

Kindly address all communications regarding this application to:

Jonathan Velasco
Sierra Patent Group, Ltd.
P.O. Box 6149
Stateline, Nevada 89449
Telephone: (775) 586-9500
Fax: (775) 586-9550

No fee is being paid at this time.

Respectfully submitted,
Sierra Patent Group, Ltd.

Dated: July 19, 2000

Sierra Patent Group, Ltd.
P.O. Box 6149
Stateline, NV 89449
(775) 586-9500

Jonathan Velasco
Reg. No.: 42,200

This application is submitted in the names of inventors Edgar Allan Tu and Eric Pang, assignor to Monggo, Inc, a California corporation.

SPECIFICATION

METHOD AND APPARATUS FOR A SECURE REMOTE ACCESS SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention pertains generally to remote access systems. More particularly, the invention is a method and apparatus which provides for remote and secure access to a host computer, and which further provides an open application standard for client access to the host computer.

2. The Prior Art

In general, remote access systems allow a "remote" user (from a remote computer) to connect to and access resources on another computer. For example, a user on a mobile computer may connect to and access resources on a home computer via conventional remote access systems. However, prior art remote access systems require special application software to be supplied to both the

remote system and the base system. Due to this shortcoming, most prior art remote access systems are limited to devices including substantial computing capabilities in the remote computer. Also, access to another computer via a remote access system is provided using conventional data connection means, typically through a PSTN (public switched telephone network) connection. That is, a direct connection from the remote computer to the base computer is typically required for security reasons.

Remote access systems can generally be categorized into two types of systems. The first system is generally referred to as a remote access server (RAS) system. A RAS system usually comprises server RAS software residing on a RAS server and client RAS software residing on a "remote" computer. The RAS server is coupled to resources (e.g., printers, files, other nodes) which are remotely accessed by a user of the system. In operation, a user of the remote computer connects to the RAS server via a dial-in telephone connection. Upon connection, the RAS server queries for the user's access credentials (e.g., user name and password). Upon authentication of the user's access credentials, the user is granted access to resources on the RAS server and/or resources on other nodes connected to the RAS server to which the user is authorized access. The RAS software manages the connection process, the authentication process, the access privileges, and the data transfers between the RAS server and the remote computer. RAS systems are also used by commercial service providers, such as Internet Access Providers (ISPs) to allow their customers access into their network resources.

In another implementation, RAS systems may be used in conjunction with an Internet connection. In this scheme, a user is able to access a RAS server indirectly via the Internet, rather than directly via a point to point telephone connection. These RAS systems are generally referred to as virtual private networks (VPNs), because a secure channel is provided via the normally unsecured Internet. In VPNs, a remote user having a computer operatively coupled to the VPN, is able to access resources on another computer via the Internet using Internet protocols.

The other type of remote access system is generally referred to as a remote control system (RCS). RCSs allow a remote user to not only access resources on another "host" computer, but also allow the user to control the host computer. RCSs typically display on the remote computer what would normally be displayed on the host computer (known as screen emulation). In this way, the user is able to control the host computer from the remote computer as if the user was directly accessing the host computer. An example of a commercially available RCS product is PC Anywhere™ by Symantec Corp.™. Like RAS systems, RCS allows a remote user to connect via a conventional means, including a telephone connection and via the Internet. Again, special software is required on both nodes.

There are several disadvantages with RAS and RCS systems. In RAS systems, file synchronization poses a common problem, particularly with respect to email applications. For example, where a remote user downloads email to the remote computer it may be stored on the remote computer. Thus, when the user

gets back to the local computer, that email is not accessible on the remote computer, but must somehow be transferred from the remote computer or disregarded. This can become quite frustrating to the user.

In addition, in RAS implementations certain files may be unusable without the original application. For example, with certain email applications, the messages associated with the email application are commonly stored in a proprietary file format. Without the original email application, the file would be unusable to the remote user if the original application is not installed on the remote computer accessed by the user.

RCS, on the other hand, typically requires proprietary software to be installed on both the server (host) and client (remote) computers. Proprietary software limits the ability of a remote user to access the host computer, because such proprietary software may not be readily accessible.

In addition, often the setup and administration of RAS and RCS systems are cumbersome or otherwise overwhelming for the home or corporate users. Setup normally involves the assistance of a network system administrator and is usually complicated further by the fact that each user may have different remote computers and different host computers. Each setup then becomes unique and difficult.

Accordingly, there is a need for a method and apparatus which provides for remote and secure access to a host computer, and which further provides an

open application standard for client access to the host computer. The present invention satisfies these needs, as well as others, and generally overcomes the deficiencies found in the background art.

BRIEF DESCRIPTION OF THE INVENTION

The present invention is a secure remote access system and method for providing access to a host computer or base appliance. The invention further relates to machine readable media on which are stored embodiments of the present invention. It is contemplated that any media suitable for retrieving instructions is within the scope of the present invention. By way of example, such media may take the form of magnetic, optical, or semiconductor media. The invention also relates to data structures that contain embodiments of the present invention, and to the transmission of data structures containing embodiments of the present invention.

The remote access system comprises a web server services module and one or more "user" server services modules. The web server services module and the user server services module of the present invention may be executed on a single data processing means or computer, but are preferably executed on a plurality of computers and according to the load balancing algorithm of the present invention as described more fully below.

The web server services module executes on a server device (web server) and is operatively coupled to a remote access device accessed by a user of the system. The remote access device may be any data processing means configured to execute web browsing software. When the user establishes a connection from the remote access device to the web server, the web server determines what type of device the user is accessing. The present invention supports a plurality of remote access devices including, for example, a computer, a mobile telephone, a personal digital assistant (PDA), or other Internet appliance device, such as a set-top web terminal (e.g., WebTV™).

A load balancing module operates in conjunction with the web server to determine to which user server module the user is redirected. The load balancing algorithm is described more fully below. In general, user servers are delegated on a "user" basis rather than on a request basis. The user of the remote access device is then redirected to a user server module according to the type of remote access device and according to the load balancing module. Further transactions between the user and the remote access system are carried out by the delegated user server module.

Each user server services module operates on a conventional computer. In order to provide robust performance, a plurality of servers (server farm) are provided for each user type, although as noted before, the invention may be carried out on a single computer. For example, a "computer" server farm is defined for computer users, a "mobile phone" server farm is defined for mobile phone users, a "PDA" server farm is defined for PDA users, and an "Internet

appliance" server farm is defined for Internet appliance users. Other server farms may be further defined for other device users. As noted above, the user of the remote access device is redirected to one of the user servers according to the type of the remote access device and according to the load balancing scheme.

The user server modules are further coupled to one or more base devices. Each base device is identified with a particular user of the system. Thus for each user accessing the system via the remote access device, a corresponding base device is identified with the user. For example, a user may be using a cellular phone (the remote access device) via the remote access system to access address book entries on the user's home computer (the base device).

The invention provides means to securely communicate data between the base machine and the user of the remote access device. In general, the communication between the user server module and the remote access device is a secure channel connection using hypertext transfer protocol. In this way, a conventional browser on the remote access device is suitable for use with the present invention without requiring proprietary software.

The invention provides a second secure channel between the user server module and the base device using a private protocol. To provide additional security, the invention provides that data communications between the user server module and the base device be initiated by the base device, rather than the server module. The details for carrying out communication transactions with the base device is described more fully below. In general, the base device initiates a

request to the user server module which opens a communication socket between the base device and the user server module. In particular, the communication socket permits the base device and the user server module to communicate through a firewall. Once this communication socket is open, the server module is able to issue commands to the base device. In response, the base device then executes the command.

Co-pending application entitled REMOTE ACCESS COMMUNICATION ARCHITECTURE APPRATUS AND METHOD, filed July 19, 2000, having attorney docket number MONG-00-001, which is expressly incorporated herein by reference, describes a system architecture and method suitable for use with the present invention.

An object of the invention is to provide a remote access system and method which overcomes the deficiencies of the prior art.

Another object of the invention is to provide a remote access system and method which provides an open application standard for client access to the host computer

Further objects and advantages of the invention will be brought out in the following portions of the specification, wherein the detailed description is for the purpose of fully disclosing the preferred embodiment of the invention without placing limitations thereon.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be more fully understood by reference to the following drawings, which are for illustrative purposes only.

FIG. 1 is a functional block diagram showing a remote access system in accordance with the present invention.

FIG. 2 is a functional block diagram showing an account creation server module in accordance with the present invention.

FIG. 3 is a functional block diagram showing a web server module in accordance with the present invention.

FIG. 4 is a functional block diagram showing a user server module in accordance with the present invention.

FIG. 5 is a flow chart showing generally the acts associated with registering a user in accordance with the present invention.

FIG. 6 is a flow chart showing generally the acts associated with redirecting a remote access user to a user server module in accordance with the present invention.

FIG. 7 is a flow chart showing generally the acts associated with communicated data between a remote access device and a base device in accordance with the present invention.

FIG. 8 is a flow chart showing generally the acts associated with a segmented read sequence in accordance with the present invention.

FIG. 9 is a flow chart showing generally the acts associated with a write sequence in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Persons of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons having the benefit of this disclosure.

Referring more specifically to the drawings, for illustrative purposes the present invention is embodied in the apparatus shown FIG. 1 through FIG. 4 and the method outlined in FIG. 5 through FIG. 9. It will be appreciated that the apparatus may vary as to configuration and as to details of the parts, and that the method may vary as to details and the order of the steps, without departing from the basic concepts as disclosed herein. The invention is disclosed generally in terms of a remote access system and method, although numerous other uses for the invention will suggest themselves to persons of ordinary skill in the art.

Referring first to FIG. 1, there is shown a block diagram of a remote access system 10 suitable for use with the present invention. The system 10 comprises a main web server 12, a database server 14, an account creation server 16, and a plurality of user server modules (generally designated 18) each coupled to the other devices 12, 14, 16, 18 via a network connection 20.

The user server modules 18 comprises a plurality of "user" servers, each identified with a remote user type. In the example system 10 depicted in FIG. 1, a computer user server 22 is provided for conventional computer users, a mobile phone user server 24 is provided for mobile phone users, a PDA user server 26 is provided for personal digital assistant (PDA) users, and an Internet appliance user server 28 is provided for other conventional Internet appliance users. Other "user" server modules may be further implemented to provide access to a particular remote user type as the need arises.

A Sili server 30 is further provided in system 10 and is coupled to the network 20 for communication to the user server modules 18. Server 12, 14, 16, 22, 24, 26, 28, 30 may be any standard data processing means, including a minicomputer, a microcomputer, a UNIX® machine, a mainframe, a personal computer (PC) such as an INTEL® based processing computer or clone thereof, an Apple® computer or clone thereof, or a SUN® workstation or server, or other appropriate computer. As such, servers 12, 14, 16, 22, 24, 26, 28, 30 generally include conventional hardware components (not shown) such as a motherboard, central processing unit (CPU), random access memory (RAM), hard disk drive,

display adapter, other storage media, a monitor, keyboard, mouse, and other user interface means, a network interface card (NIC), and/or other conventional input/output devices.

Each server 12, 14, 16, 22, 24, 26, 28, 30 has loaded in its RAM conventional operating system software (not shown) such as UNIX®, Linux™, Windows NT®, Novell®, Solaris® or other server operating system. Main Web Server 12 further has loaded in its RAM the web server software (not shown) for handling http (hyper text transfer protocol) or web page requests from remote users. Web Server 12 is described more fully below in conjunction with FIG. 3 and FIG. 6.

Database server 14 further has loaded in its RAM conventional database software (not shown) such as Oracle®, IBM® DB2, Microsoft SQL® or other appropriate database software for storage, update and retrieval of system user data, and as described more fully below in the operation of the user server modules (FIG. 7). The account creation server 16 carries out the operation of defining new system user accounts and is described more fully below in conjunction with FIG. 2 and FIG. 5.

Each of servers 12, 14, 16, 22, 24, 26, 28, 30 may further comprise a server farm, rather than a single server as is known in the art. For example, computer user server 22 may comprise a cluster (or plurality) of computer user servers. In the preferred embodiment, each of the user server modules 22, 24, 26, 28 comprise server farms. Where each user server module is structured and configured as a

server farm, the system further provides a load balancing module (depicted in FIG. 3) in the main web server 12. The load balancing module of the invention is described in more detail below in conjunction with FIG. 3 and FIG. 6. The main web server 12 may also comprise a server farm to provide robustness to the system 10 when a plurality of remote access devices are accessing the system 10 concurrently.

The system's network connection 20 is operatively coupled to an Internet terminal adapter 32 for providing access to the global information network, known as the Internet 36. The terminal adapter 32 provides a serial connection to an Internet Service Provider (ISP) 34 or other Internet backbone and couples the system 10 to the Internet 36.

The system 10 may be operatively coupled with one or more remote access devices (not shown) via the Internet 36. Such remote access devices may be conventional computers (including desktop, portable notebooks, and palmtop computers), mobile telephone devices (such as cellular and PCS (personal communications service) phones), personal digital assistants (such as Palm Pilot™ or Windows CE™), and other Internet appliance devices (such as WebTV™). In general, a remote access device which may communicate over the Internet and is capable or viewing Web pages is suitable of communication with the system 10.

The system 10 may further be operatively coupled with one or more "base" devices (not shown) via the Internet 36. The base devices typically connect to the Internet using conventional connection means, such as dial-up, cable, or

network connections, for example. Each base device contains or provides an access gateway to information which is provided to the user of the remote access device. Such information may include, for example, computer files such as address book files, document files, email documents, among others. Each base device is identified with a user of the remote access device via conventional authentication means, such as challenge and response authentication. For example, when a remote user provides a user name and password to the system 10, the system 10 then identifies the base device which the user is entitled to access.

The base devices may be any conventional data processing means or computer suitable for communicating data to the user server modules 18 in accordance with the present invention and as described in further detail below (FIG. 7 through FIG. 9). The "base" device and its operation is described in copending application entitled " AGENT SYSTEM FOR A SECURE REMOTE ACCESS SYSTEM " having the attorney docket number MONG-00-003 and filed July 19, 2000, the disclosure of which is expressly incorporated herein by reference.

In cases where a base device does not have a permanent (i.e., persistent or "full-time") connection to the Internet, the Sili server 30 is configured to "wake-up" the base device. Normally, this process is carried out when a user identified with the base devices is accessing the system 10 via a remote access device. Accordingly, the system 10 may be further coupled to the base devices via the Sili server 30. The Sili server 30 is coupled to a modem pool 38 which may comprise

a bank or pool of modems as is known in the art. The Sili server 30 is configured to dial the base device by calling a phone number designated for the base device via PSTN (public switched telephone network) 40 and negotiate a connection between the base device and Sili server 40 via the Internet. During negotiation, the Sili server 30 typically identifies the identity (or location such as the IP address) of the Sili server 30, and then terminates the PSTN 40 connection. In response, the base device then carries out the operation of connecting to the Internet and communicating with the Sili server 30 over the Internet connection. Once connected with the Sili server 30 via the Internet (whether permanently or as requested during the above described "wake up" process), the Sili server 30 then communicate which user server module is requesting data from the base device. Further processing is carried out between the designated user server module and the base device as described below in conjunction with FIG. 4 and FIG. 7.

Referring next to FIG. 2, as well as FIG. 1, there is shown a block diagram of the account creation server 16 according to the present invention. The account creation server 16 includes an agent communication module 44 operatively coupled to a user registration module 46, each operating in RAM. The agent communication module 44 is further coupled for communication to the base device 42 via conventional connection means, such as via the Internet or a modem connection. In general, a user of the system registers to provide access to information on (or coupled to) the base device 42. For example, a user may register to provide access to a home computer (base device). Another example is where a corporate user registers to provide access to one or more office

computers. After registration, the user is able to access data on the base device via various remote access devices.

The account agent communication module operates using user data provided by the base device 42. During registration of a user, the base device 42 communicates to the account creation server 16 the user's name, password, IP address of the base device, modem phone number of the base, among other user data such as name, sex, birthday, email, email password, email client, zip code, for example. The agent communication module 44 receives this user data and provides the user data to the user registration module 46 for further processing. The user registration module 46 receives the user data and creates an "account" for the user by storing the user data into the database server 14. When the user later accesses the system 10 via a remote access device, the user's credential may be verified according to the user's name and password, and the IP address and/or modem phone number of the base device identified with the user may be ascertained from the database 14. Access to information on (or coupled to) the base device may thereafter be provided to the user of the remote access device.

Referring now to FIG. 3, as well as FIG. 1 and FIG. 2, there is shown a block diagram of a main web server 12 in accordance with the present invention. As noted above, the tasks carried out by the web server 12 may be carried out by a server farm comprising a plurality of web servers, each configured substantially as described herein for main web server 12.

Main web server 12 comprises a welcome handler module 48 coupled to a user server module redirector module 50, which is further coupled to a load balancing module 52. Each of the modules 48 through 52 operates within the RAM of the web server 12 to carry out the operations described herein.

The main web server 12 is operatively coupled to one or more remote access devices 54 (via the Internet) which are accessed by a user (remote user) of the system 10. As described above, the remote access device 54 may be any data processing means suitable for connecting to the Internet and viewing web pages. For example, the remote access device 54 may be a computer, mobile phone, PDA, or other Internet appliance device.

A user of the remote access device 54 typically initiates communication with the system 10 by connecting to the main web server 12 and requesting a web page (via conventional http requests). The welcome handler 48 receives this user request and determines, among other things, the type of device the user is using. This determination is typically carried out by inspecting the request string communicated by the remote access device 54 and determining the browser type field in the request string. In general, the browser type field identifies the browser version including the hardware platform (i.e., the device type). Using the data in the request string, the welcome handler 48 identifies the device type and communicates the device type to the user server module redirector module 50 for further processing.

The user server module redirector 50 receives the device type from the welcome handler 48 and then queries the load balancing module 52 to determine the appropriate user server module 22 through 28 to which the remote user is redirected.

The load balancing module 52 carries out an enhanced user server module designation algorithm which overcomes disadvantages in the prior art. Prior art load balancing allocates resources on a request by request basis without regard to the who is making the request, and therefore resources of a plurality of machines may be delegated to a single user. In contrast, the load balancing module 52 allocates user server module designation on a user by user basis, rather than based on requests. Thus, a user is designated a particular user server module 18. Requests made by the same user are directed to the same user server module during a "session" defined for the user.

A session is defined for a user when the user initiates a first request the system 10. The session remains active and further requests are identified with the same session, until either the user signs out (logs off) or a predefined period of inactivity has elapsed, such as when the user has not made any requests within a predetermined timeout period (e.g., 30 minutes).

According to this algorithm, the load balancing module 52 designates a user server module within each server farm according to usage on a session (or user) basis, rather than on a request basis. If, for example, the remote access device type is determined by the welcome handler 48 to be a mobile phone, the

load balancing module 52 designates a mobile phone user server 24 from the servers in the mobile phone user server farm according to session usage (or user usage). This designation is then communicated to the user server module redirector 50.

The user server module redirector 50 receives the user server module designation from the load balancing module 52 and transmits a command (typically an "http redirect" command) to the remote access device 54 which redirects the remote access device 54 to the designated user server module.

Referring now to FIG. 4, as well as FIG. 1 through FIG. 3, there is shown a block diagram of a user server module 18 according to the present invention. User server modules 22 through 28 are configured substantially as user server module 18 described herein. User server modules 22 through 28 differ from each other in that the data formatted communicated to the remote access device according to the remote access device type and as defined in the document templates described below. In its most broadest description, the user server module 18 provides a user of a remote access device to access data on a base device via an open standard remote access platform such a web browser. The user server module 18 also provides means for formatting data according to the device type of the remote access device. In addition, the user server module 18 provides a secure means for retrieving requested information from the base device via requests which are initiated by the base device, rather than by the user server module 18.

The user server module 18 comprises a user request handler module 56, a data parser/formatter module 58, an agent communication module 60, a plurality of document templates 62 and a user data module 64. The user request handler module 56 is operatively coupled to the remote access device 54 via the Internet and the database server via conventional networking means. The user request handler module 56 is also coupled for communication to the data parser/formatter module 58. The user request handler 56 carries out the operation of receiving and responding to requests from users (accessing the remote access devices 54). The user request handler 56 also verifies the users access credentials using conventional authentication means, such as challenge and response authentication. For example, in the illustrative example module 18, the user request handler 56 verifies the user's access credentials by querying the database server 14 and verifying the user's name and password. This verification is normally carried out when the user is first redirected to the user server module 18 by the main web server 12. Once this verification is established a "session" is maintained for the user, as described above.

The user request handler 56 also forwards user requests to the data parser/formatter 58 (for further processing) and receives formatted documents from the data parser/formatter 58 for communication to the remote user at the remote access device 54. The communication between the user request handler 56 and the remote access device 54 may be carried out over a secure connection, such as secure sockets layer, but may also carried out over a conventional http connection.

The data parser/formatter module 58 is further coupled to the agent communication module 60, the user data module 64 and the set of document templates 62. The data parser/formatter 58 receives user requests from the user request handler 56. User requests typically comprise requests to view, read, write, modify, or delete data on a base device 42, which is operatively coupled to the user server module 18 via the agent communication module 60. The base computer 42 is identified with the user of the remote access device 54 during the authentication of the user, using information provided by the user during registration (as described above in conjunction with FIG. 2).

When the data parser/formatter 58 receives a particular user request, the requested information may be in the user data module 64, which is a conventional data storage facility maintained during the current session of the user. Data may be in the user data module 64 if the same information had been previously requested in the same session. However, data in the data module 64 is purged when the session is terminated (which may occur upon the user log out or upon the expiration of a predetermined session timeout period). If the data requested is not in the user data module 64, the data parser/formatter 58 requests the data from the base computer 42 via the agent communication module as described further below. The requested data is then provided from the agent communication module 60 to data parser/formatter 58, which stores the data in the user data module 64 for prospective requests.

The data parser/formatter 58 then formats the requested data (either obtained from the user data module 64 or from the base device 42 via the agent

communication module 60) and merges the data with one of the document templates 62. The document templates 62 are provided in different "formats" according to the remote access device type and according to the request type. For example, the documents templates 62 for the computer user server 22 provides document formats appropriate for viewing on a conventional computer device (i.e., more complex objects such as frames, images, scripting, sound, for example). Likewise, the documents templates 62 for the mobile phone user server 24 provides document formats suitable for viewing on a mobile phone device (e.g., fewer complex objects, such as primarily text-oriented formats).

Additionally, different formats of document templates 62 are provided according to the request type (and the type of content to view viewed). For example, if the user is requesting to view the user's e-mail inbox, the appropriate objects and formats are provided to facilitate viewing of the user's e-mail inbox (e.g., icons for the address book, icons for messages, icons for sending a new message). If, on the other hand, the user is requesting to view the user's drive directory, the appropriate objects and formats are provided to facilitate view of the user's directory structure (e.g., folder icons to represent directories, expanding tree objects to view sub-directories). Other formats and objects are further defined in the document templates 62 to provide viewing of data according to the requesting device type.

As depicted in FIG. 4, the agent communication module 60 is further coupled for communication to the Sili server 30 and configured to be coupled with the base device 42. For example, when the agent communication module 60

receives a request for data in a base device 42, the agent communication module 60 transmits a request ("connection request") to the Sili Server 30 for the base device 42 to the connect to the particular user module 18. In response the Sili Server 30 communicates to the base device that the user module 18 has a pending job request as described herein.

The Sili server 30 is configured to be coupled with the base device 42 via the Internet (although the invention may be also be carried out where the base device 42 and the Sili server 30 are coupled using other connection means, such as a direct serial connection, for example). In the case where the base device 42 maintains a "full time" (or permanent) connection to the Internet, the Sili server 30 and the base device 42 periodically communicate signals back and forth.

Preferably, communications between the base device 42 and the Sili server 30 are initiated by the base device 42. For example, a base device 42 which maintains a full time Internet connection is generally configured to periodically communicate "job request" commands at a predetermined interval (e.g., forty (40) seconds) to the Sili server 30. In response, the Sili server 30 may indicate "no job" or "job request by a user server module". "No job" is communicated where the user associated with the base device 42 is not requesting data at this time. "Job request by a user sever module" is communicated when the user associated by the base device 42 is requesting data (which is indicated to the Sili server 30 by the agent communication module 60 as noted above). Where the Sili server 30 indicates that a job request is pending, the Sili server 30 also identifies the

particular user server module (22 through 28), normally by identifying the IP address of the particular user server module.

If the base device 42 does not maintain a full-time Internet connection, further processing may be required to establish a communication between the Sili server 30 and the base device 42 over the Internet. The Sili server 30 may readily determine whether a particular base device 42 maintains a full time Internet connection by checking whether the base device 42 communicates period "job request" commands as described above. Where the base device 42 does not maintain a full-time connection to the Internet, the invention provides means for signaling the base device 42 to establish an Internet connection and connect to the Sili server 30.

In the present illustrative embodiment shown in FIG. 1 and FIG. 4, the Sili server 30 comprises a "wake up" module 66 which operates in RAM. The Sili server 30 is further coupled to a modem pool 38 having means for connection via the PSTN 40. When the Sili server 30 receives a connection request from a user server module 18 for a particular base device 42 which does not maintain a full-time Internet connection, the wake up module 66 connects to the base device 42 by dialing the phone number of the base device 42 through the modem pool 38 (and PSTN 40) and indicating that connection to the Sili server 30 via the Internet is requested. The phone number of the base device 42 may be readily obtained from the database server 14. If necessary, the IP address of the Sili server 30 may also be conveyed to the base device 42. After communicating this information to the base device 42, the Sili server 30 terminates its connection via

the modem pool 38. Responsive to the wake up command from the Sili server 30, the base device 42 establishes a conventional connection to the Internet (normally via an ISP) and thereafter communicates job request commands to the Sili server 30.

Referring again to the agent communication module 60 in FIG. 4, when the agent communication module 60 receives a request for data in a base device 42, the agent communication module 60 transmits a request ("connection request") to the Sili Server 30 for the base device 42 to the connect to the particular user module 18, as noted above. In response, the Sili Server 30 communicates to the base device 42 that the user module 18 has a pending job request, by responding to a "job request" command from the base device 42 with a "job request by a user server module" response and indicating the IP address of the particular user server module. The base device 42 receives the "job request by a user server module" response and thereafter communicates with specified user server module while the associated "session" is active. Once, the session is terminated, the base device 42 either reestablishes its connection with the Sili server 30 (communicating job request commands) or terminates its connection with the system 10 and the Internet. The base device 42 reestablishes its connection with the Sili server 30 at the end of the session if the base device 42 is configured for a full-time Internet connection. Otherwise, it terminates its connection with system 10 and the Internet.

When the base device 42 receives the "job request by a user server module" response from the Sili server 30, it begins communication with specified

user server module, and more particularly the agent communication module 60 in the specified user server module. Preferably, communication between the agent communication module 60 and the base device 42 is initiated by the base device 42, using a task connection request. For example, the preferred sequence of communication comprises a task connection request communicated from the base device 42 to the agent communication module 60. This task connection request communication opens a communication socket between the base device 42 and the agent communication module 60. If the agent communication module 60 has pending requests (e.g., received from the data parser/formatter 58) the agent communication module 60 communicates a task command reply in response to he task connection request. The task command reply may indicate the data requested from the base device 42, such a directory listing, or a transaction (e.g., send or delete), for example. Responsive to the task command reply, the base device 42 provides the requested information to the agent communication module 60 or carries out the requested instruction. Additional request may be carried out by the agent communication module 60 by transmitting further task commands to the base device using the open communication socket.

As described above, communication sequences between the system 10 (Sili server 30 and user server module 18) and the base device 42 are generally initiated by the base device 42, rather than the system 10. The exception is where the Sili server 30 initiates a wake-up sequence as described above. However, for data transfers and key operations (such as file transaction), communications are initiated by the remote device. This arrangement provides several advantages which overcomes problems associated with the prior art. First, security is

increased since the data communications are initiated by the base device rather than by the system 10. By requiring the base device to initiate communication (and therefore establish a connection socket), hacking into the base device from the outside becomes a more difficult task. Additionally, the invention may be practiced even if the base device is behind firewall because the base device initiates communication and opens the connection to the agent communication module, thereby allowing reply communications and task commands to be communicated from the agent communication module.

The communication between the base device 42 and the system 10 (sili server 30 and agent communication module 60) are preferably carried out over a secure connection utilizing for example, 128-bit encryption. Additionally, a private (non-public) communication may be provided by the system as a communication means between the system 10 and the base device 42 as is known in the art.

As noted above, the data parser/formatter 58 carries out the operation of requesting data from the base device 42 via the agent communication module 60 and storing "session" data in the user data module 64 during the active session. Where the data requested from the base device 42 is stored in a substantially large file (e.g., greater than one megabyte), the data parser/formatter 58 further carries out the operation of segmenting the file and requesting only the portion of the file having the requested data. For example, e-mail mailbox files generally grow in size over time. The user, however, may only be requesting one particular message, rather than all the messages in the mailbox. Accordingly, it would be

inefficient to transfer the entire mailbox file, when only a portion having the requested message is requested. The details of carrying out segmentation with the base device 42 are described more fully below in conjunction with FIG. 8 and FIG. 9.

The method and operation of invention will be more fully understood with reference to the flow charts of FIG.5 through FIG. 9, as well as FIG. 1 through FIG. 4. The order of actions as shown in FIG. 5 through FIG. 9 and described below is only exemplary, and should not be considered limiting.

Referring now to FIG. 5, as well as FIG. 1 through FIG. 4, the acts associated with registering a user in accordance with the present invention are generally shown. In general registration of a user is communicated by the user from the base device 42, wherein the user is establishing a user name, password, the information (e.g., phone number, IP address) about the base device which the user is enabling for remote access.

At box 100, the account creation server 16 receives the registration data from the base device 42. The registration data includes such information as user name, password, base device phone number, base device IP address, which identifies the user (via user name and password) with the particular base device (via phone number and/or IP address). Box 110 is then carried out.

At box 110, the account creation server 16 stores the registration data received in box 100 into the database server 14 using conventional database

commands as is known in the art. The registration data, once stored in the database server 14, may be accessed to verify a user attempting to remotely access the base device.

Referring now to FIG. 6, as well as FIG. 1 through FIG. 5, the acts associated with redirecting a remote access user to a user server module 22 through 28 in accordance with the present invention are generally shown. Remote users generally access system 10 via the Internet 36 and a remote access device capable of connecting to the Internet and viewing web pages (such as a web browser). The remote users initiate communication with the system 10 by transmitting a conventional http request to the main web server 12 via the URL address of the main web server 12.

At box 200, the welcome handler 48 in the main web server 12 receives the http (initial entry) request transmitted from the remote device by the remote user. Box 210 is then carried out.

At box 210, the welcome handler 48 determines the type of remote access device the remote user is accessing. As described above, this determination is typically carried out by inspecting the request string communicated by the remote access device 54 and determining the browser type field in the request string. In general, the browser type field identifies the browser version including the hardware platform (i.e., the device type). Using the data in the request string, the welcome handler 48 identifies the device type and communicates the device type

to the user server module redirector module 50 for further processing. Diamond 220 is then carried out.

At diamond 220, the user server module redirector 50 receives the device type from the welcome handler 48 and then queries the load balancing module 52 to determine the appropriate user server module 22 through 28 to which the remote user is redirected. The load balancing module 52 determines whether an session is currently active for the particular user by inspecting a local data record of sessions. If a session is currently active for the user, box 230 is then carried out. Otherwise, Box 240 is carried out.

At box 230, the current user is already associated with an active session (and user server module). Box 250 is then carried out to redirect the user to the appropriate user server module.

At box 240, the current user is not associated with an active or existing session. The load balancing module 52 assigns a session to the current user and designates a user server module to the session according to the remote device type and the current allocation of sessions within the user server module farm for the remote device type. For example, if the remote device type is determined to be a mobile phone, the load balancing module 52 assigns a user server from the mobile phone user server farm 24. The server in the mobile phone user server farm 24 which has the least assigned sessions is generally designated the current session. Box 250 is then carried out.

At box 250, the user server module redirector 50 redirects the remote user (via http commands) to the designated server. As described further below, subsequent requests from the remote user to the system 10 will be directed to the designated server while the current session remains active. Process 260 is then carried out.

At process 260, communication sequence between the designated user server module and the remote device begins. This process is described more fully below in conjunction with FIG. 7.

Referring now to FIG. 7, as well as FIG. 1 through FIG. 6, the acts associated with communicated data between a remote access device and a base device by the system 10 in accordance with the present invention are generally shown. This process is carried out after the remote access device is designated a user server module as described above for FIG. 6.

At box 300, the designated user server module (generally designated as 18) process begins, normally after URL (or IP) redirection by the main web server 12. After initial redirection, the user request handler 56 in the user server module 18 provides a web page (login screen) requesting the remote user's access credentials (normally a user name and password). The format/layout for the login screen may be provided by the data parser/formatter 58 and document templates 62 or may alternatively be a standard document produced by the user request handler 56. Box 310 is then carried out.

At box 310, the user request handler 56 receives the access credential's of the remote user from the remote access device and verifies the information by querying the database server 14 using conventional database requests. Diamond 320 is then carried out.

At diamond 320, the user request handler 56 determines whether the access credentials provided by remote user matches information in the database server. If so, the user is authenticated and box 340 is carried out. Otherwise, the user was not verified and box 330 is carried out.

At box 330 the user's access credentials were not verified. The user request handler 56 may provide additional attempts for the user to provide the correct access credentials by providing another login screen. Box 310 is then repeated.

At box 340, the user's access credentials were authenticated. The user request handler 56 refreshes the remote user's screen by transmitting a "main" or "menu" page providing the user with a plurality of options, such as transmitting or viewing a file, for example. Diamond 350 is then carried out.

At diamond 350, the agent communication module 60 communicates with the Sili server 30 to determine whether base device corresponding to the current user is currently connected to the system 10 via the Internet. As noted above, some base device may have full-time Internet connections, in which case the base device is periodically communicating with the Sili server 30 as described above

(see FIG.4). Since a session is currently active, the Sili server 30 communicates a "job request for user server module" command indicating the address of the designated user server module 18 to the base device in response to a "job request" transmission from the base device.

Base devices which do not have full-time Internet connections must be signaled to connect to the system 10 (also described above in conjunction with FIG. 4). If the base device is currently communicating with the Sili server 30, box 370 is carried out. Otherwise box 360 is carried out.

At box 360, the "wake-up" sequence is initiated wherein the Sili server 30 dials the designated number for the base device via the PSTN 40 as described above in FIG. 4. The Sili server 30, among other things, requests the base device to connect to the Internet and thereafter connect to the Sili server 30. The Sili server 30 then terminates its PSTN 40 connection. In response, the base device establishes an Internet connection (normally via a ISP) and thereafter connects to the Sili server 30. After the base device and the Sili server commence communication. The Sili server 30 communicates a "job request for user server module" command indicating the address of the designated user server module 18 to the base device in response to a "job request" transmission from the base device. Box 370 is then carried out.

At box 370, the user request handler 56 waits for a task command from the remote user, which is normally communicated from the remote access device to the user server module 18 via conventional http protocols, such as selecting an

option from the web page provided by the user request handler 56 to the remote access device. As noted above, the communication between the remote access device and the user server module 18 may be a secure transaction with encryption (e.g., 128-bit encryption). Box 380 is then carried out.

At box 380, the remote user has communicated a task command to the user server module 18. This request is received by the user request handler 56. If the request is for the retrieval (e.g., viewing or listing) of information, the user request handler queries the data parser/formatter 58 for the information and box 390 is carried out. If the request is to carry out or execute a command on the base device (to delete a file, for example), the request may be communicated directly via the agent communication module 60 to the base device, where the base device 42 carries out the instruction. As noted above, communication between the base device and the user server module is carried out via requests initiated by the base device. Where the request is to carry out a write (e.g., sending or changing a file on the base device) sequence, rather than read sequence, the write sequence is carried out as described in conjunction with FIG. 9.

At box 390, the agent communication module 60 awaits a task connection from the from the base device. As noted above, during the active session, the base device and the designated user server module 18 communicate via a (task connection) request initiated from the base device. Once this task connection request is received by the designated user server module 18, a socket connection is established and maintained by the user server module 18 during the active session. Thereafter, task commands may be issued by the user server module 18

to the base device. When the user server module 18 has a request pending for the base device, the user server module 18 communicates a task command reply in response to the task connection request from the base device. Accordingly, the agent communication module 60 awaits a task connection request from the base device. Box 400 is then carried out.

At box 400, the agent communication module 60 has received the task connection request form the base device (i.e., a connection socket is established) and transmits a task command. The task command indicates the requested data. Box 410 is then carried out.

At box 410, the agent communication module 60 receives from the base device task command reply data responsive to the task command issued by the user. The data is then communicated to the data parser/formatter 58 for further processing. Box 420 is then carried out.

At box 420, the requested data is stored into the user data module 64 by the data parser/formatter 58. This allows the data to be provided for subsequent requests during the active session. While resident in the user data module 64 during the active session. The data is deleted or otherwise purged when the session is terminated, as described above. Box 430 is then carried out.

At box 430, the data parser/formatter 58 merges the requested data with the appropriate document template 62 to generate a web page suitable for viewing on the remote access device. Box 440 is then carried out.

At box 440, the generated web page is communicated to the user request handler 56 for transferring to the remote access device. The web page is communicated using conventional http commands, and is provided as a reply to the task request command received from the user. Box 450 is then carried out for subsequent requests from the user.

At box 450, the user request handler 56 waits for a task command from the remote user. When a task command is received from the user, box 460 is then carried out.

At box 460, the remote user has communicated a task command to the user server module 18. This request is received by the user request handler 56. If the request is for the retrieval (e.g., viewing or listing) of information, the user request handler queries the data parser/formatter 58 for the information and diamond 470 is then carried out. If the request is to carry out or execute a command on the base device (to delete a file, for example), the request may be communicated directly via the agent communication module 60 to the base device, where the base device 42 carries out the instruction. As noted above, communication between the base device and the user server module is carried out via requests initiated by the base device. Where the request is to carry out a write (e.g., sending or changing a file on the base device) sequence, rather than read sequence, the write sequence is carried out as described in conjunction with FIG. 9.

At diamond 470, the data parser/formatter 58 determines whether the requested data is available locally (i.e., already stored in the user data module 64 from a previous transaction). If so, box 480 is carried out. Otherwise, box 400 is carried out.

At box 480, the requested data is available locally from the user data module 64. The data parser/formatter 58 merges the data with the appropriate document template 62 to generate a web page suitable for viewing on the remote access device. Box 440 is then carried out.

Referring now to FIG. 8, the acts associated with a segmented read sequence in accordance with the present invention is generally shown. This sequences may be carried out in conjunction the read sequence of FIG. 7 (Box 400 through 450) described above. Segmentation is appropriate where, for example, the requested information is only a portion of a larger file, such a where the requested information is a phone number entry in the user's address book. Other situations for segmentation will be readily apparent to those skilled in the art having the benefit of this disclosure.

At box 500, the data parser/formatter, instead of requesting the entire file, requests a file segment, and more particularly segment one (1) of the file. As is known in the art, segment 1 normally comprises such information as the data structure (index) for the file, and reference offsets to data within the file. Segment 1 is generally requested when not already resident in the user data module 64. Once segment 1 of a file is in the user data module 64, subsequent requests for

other segments (box 530 and 540 below) of the same file may be carried out without requesting segment 1. As with other requests, this task command is submitted during the active session once the connection socket is open. Box 510 is then carried out.

At box 510, the agent communication module 60 receives the requested segment 1 and communicates the file segment 1 to the data parser/formatter 58 for further processing. Box 520 is then carried out.

At box 520, the data parser/formatter 58 inspects segment 1 and stores the data locally within the user data module 64. As noted above, once segment 1 of a file is in the user data module 64, subsequent requests for other segments (box 530 and 540 below) of the same file may be carried out without requesting segment 1 from the base device. Box 530 is then carried out.

At box 530, the data parser/formatter 58 determines which data segment ("requested segment") from the original file contains the data requested by the user. The data parser/formatter then requests this agent communication module to retrieve the "requested segment". Box 540 is then carried out.

At box 540, a task command to retrieve the "requested segment" is communicated by the agent communicated module 60 to the base. Box 550 is then carried out.

At box 550, the "requested segment" is received from the base device in reply to the task command of box 540. This data in the requested segment is stored locally in the user data module 64 and web page generally is then carried out normally as described above in conjunction with box 450 of FIG. 7.

Referring now to FIG. 9, the acts associated with a write sequence in accordance with the present invention are generally shown. This sequence is generally carried when requested by the user as described above in box 380 or box 460 of FIG. 7.

At box 600, a change data request is received from the user (box 380 or box 460, FIG. 7). The request may be to update a phone book entry, for example. Box 610 is then carried out.

At box 610, the change data request is communicated by the agent communication module 60 to the base device. As with other requests, this task command is submitted during the active session once the connection socket is open. Box 620 is then carried out.

At box 620, if the data change affect information which is also stored in the user data module 64, the information in the user data module 64 is flagged as "old". Accordingly, when a subsequent request is made for the information, the read sequence with the base device is carried out again, since the information in the user data module 64 is considered outdated.

Accordingly, it will be seen that this invention provides a method and apparatus which provides for remote and secure access to a host computer, and which further provides an open application standard for client access to the host computer. Although the description above contains many specificities, these should not be construed as limiting the scope of the invention but as merely providing an illustration of the presently preferred embodiment of the invention. Thus the scope of this invention should be determined by the appended claims and their legal equivalents.

CLAIMS

What is claimed is:

1. A system for providing access to a base device identified with a user of a remote client device, said remote access system comprising:

a) a web server operatively coupled for communication with the remote client device accessed by the user; and

b) a user server operatively coupled to said web server and said remote client device, said user server further configured to communicate data between the base device and the user of the remote client device, said user server further configured to communicate data with said base device via requests initiated by said base device.

2. The remote access system of claim 1, wherein said data communicated to the remote client device is formatted for viewing by a web browser.

3. The remote access system of claim 1, wherein said data communicated to the remote device is further formatted for viewing on a personal computer.

4. The remote access system of claim 1, wherein said data communicated to the remote device is further formatted for viewing on a mobile telephone.

5. The remote access system of claim 1, wherein said data communicated to the remote device is further formatted for viewing on a personal digital assistant device.

6. The remote access system of claim 1, wherein said data communicated to the remote device is further formatted for viewing on an internet appliance device.

7. In a server device operatively coupled to at least one base device and at least one remote access device, a method for securely communicating data between the base device and the remote access device comprising:

a) authenticating a user's access credential to access the base device from the remote access device;

b) receiving a request from said user to carry out at a command on said base device;

c) awaiting a task connection request from said base device;

d) replying to said task connection request with a task connection reply to establish a socket connection;

e) communicating a command to said base device in conjunction with said task connection reply to carry out the command requested by the user;

f) receiving from said base device the results of said command; and

g) communicating to said user said results of said command.

8. The method of claim 7, further comprising communicating a wake up signal to said base device prior to awaiting a task connection request.

9. The method of claim 7, further comprising maintaining said socket connection with said base device in an open fashion and issuing further user requests via said open connection.

10. The method of claim 7, further comprising determining the device type of the remote access device and communicating information to said remote access device in a format suitable for viewing thereon according the determined device type.

11. The method of claim 10, wherein said information communicated to said remote access device is formatted for viewing by a web browser.

12. The method of claim 10, wherein said information communicated to said remote access device is formatted for viewing on a personal computer.

13. The method of claim 10, wherein said information communicated to said remote access device is formatted for viewing on a mobile telephone.

14. The method of claim 10, wherein said information communicated to said remote access device is formatted for viewing on a personal digital assistant.

15. The method of claim 10, wherein said information communicated to said remote access device is formatted for viewing on an internet appliance device.

ABSTRACT

A method and apparatus which provides for remote and secure access to a host computer, and which further provides an open application standard for client access to the host computer.

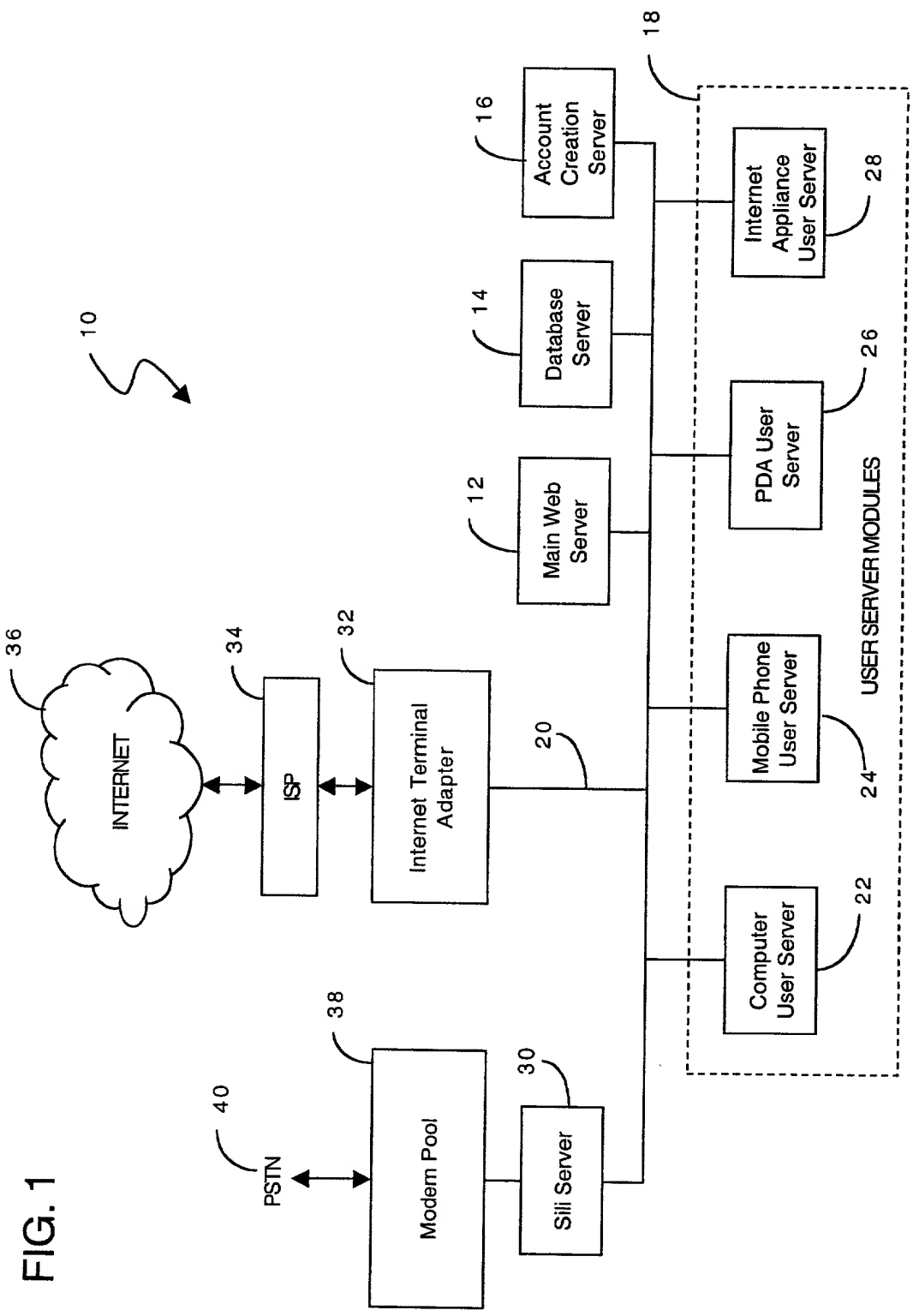

10
INTERNET
36
ISP
34
Internet Terminal Adapter
32
20
Main Web Server
12
Database Server
14
Account Creation Server
16
18
Internet Appliance User Server
28
PDA User Server
26
Mobile Phone User Server
24
Computer User Server
22
USER SERVER MODULES
Modem Pool
38
Sili Server
30
PSTN
40

FIG. 1

FIG. 2

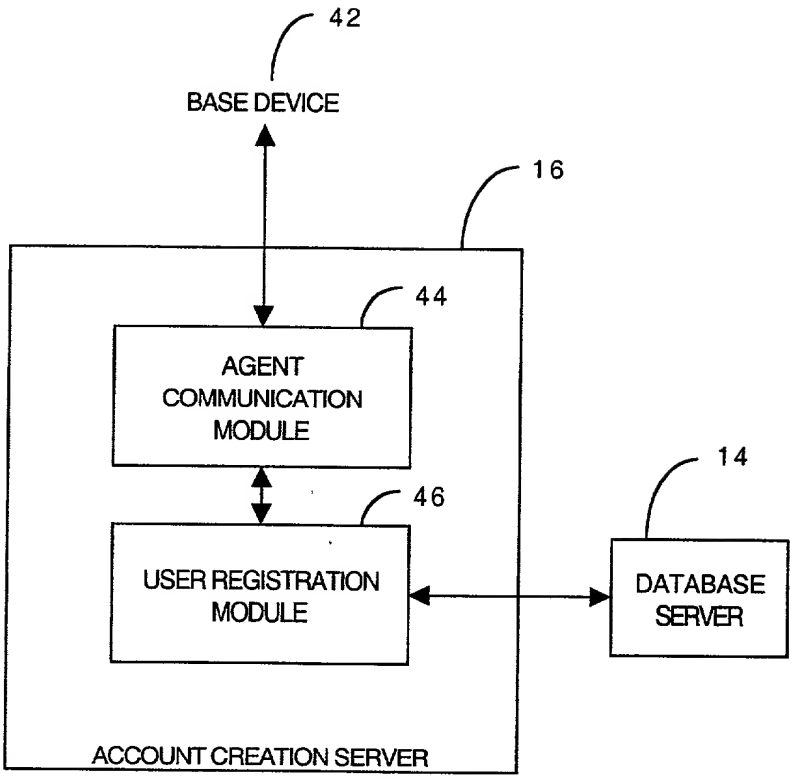
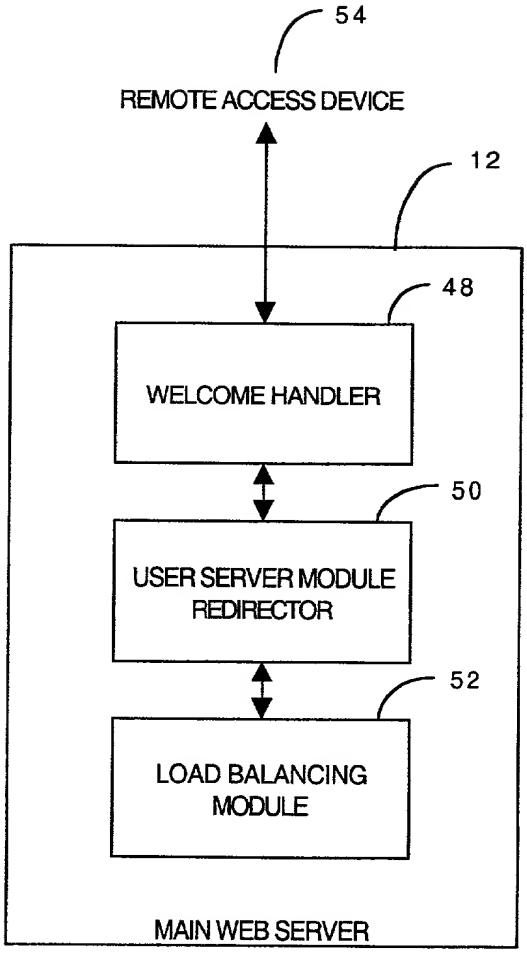

42
BASE DEVICE
16
44
AGENT COMMUNICATION MODULE
14
46
USER REGISTRATION MODULE
DATABASE SERVER
ACCOUNT CREATION SERVER

FIG. 3


54
REMOTE ACCESS DEVICE
12
48
WELCOME HANDLER
50
USER SERVER MODULE
REDIRECTOR
52
LOAD BALANCING
MODULE
MAIN WEB SERVER

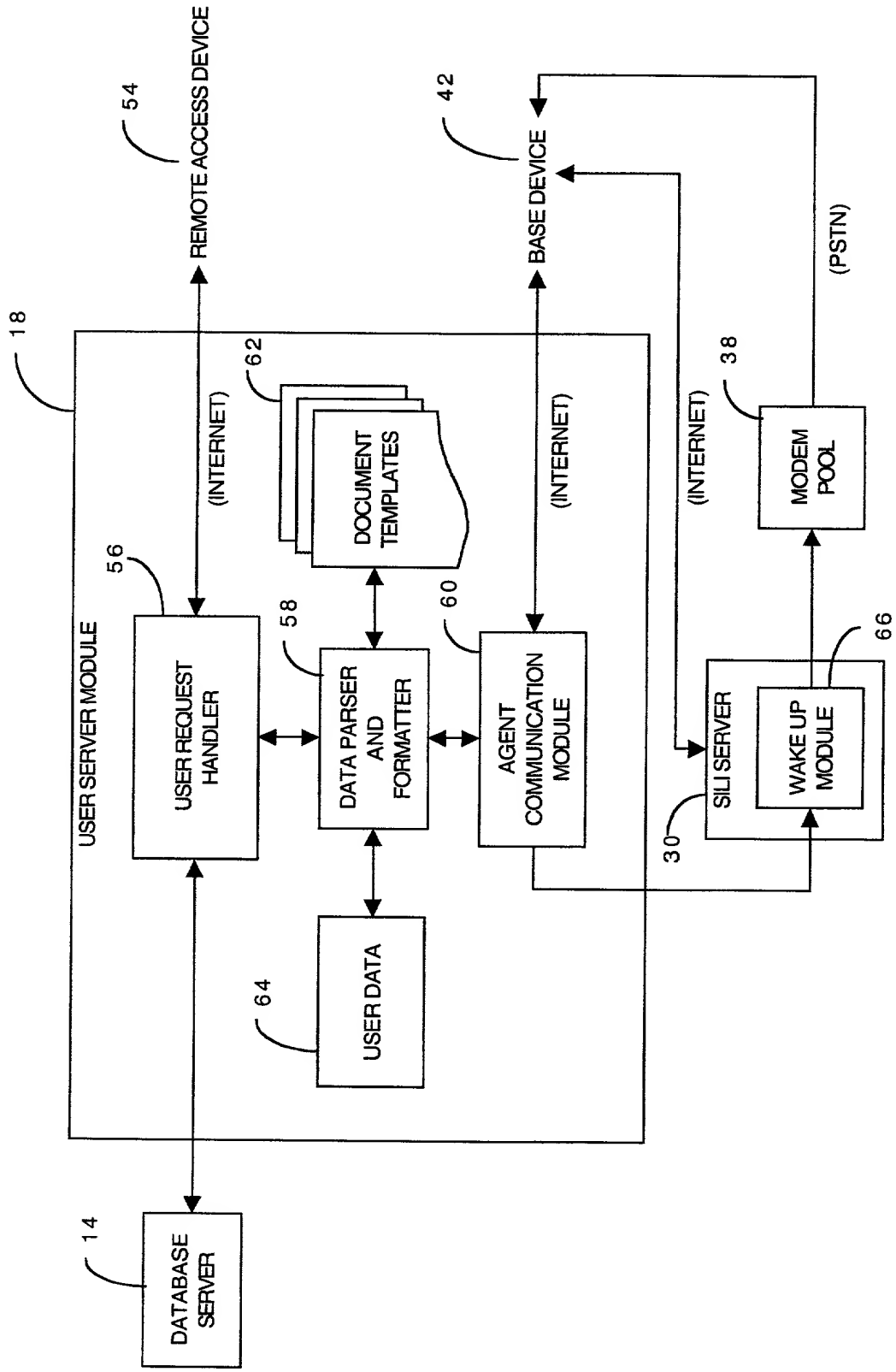

REMOTE ACCESS DEVICE
54
BASE DEVICE
42
(PSTN)
18
(INTERNET)
62
DOCUMENT TEMPLATES
(INTERNET)
(INTERNET)
38
MODEM POOL
56
58
60
66
USER SERVER MODULE
USER REQUEST HANDLER
DATA PARSER AND FORMATTER
AGENT COMMUNICATION MODULE
SILI SERVER
WAKE UP MODULE
30
64
USER DATA
14
DATABASE SERVER

FIG. 4

FIG. 5

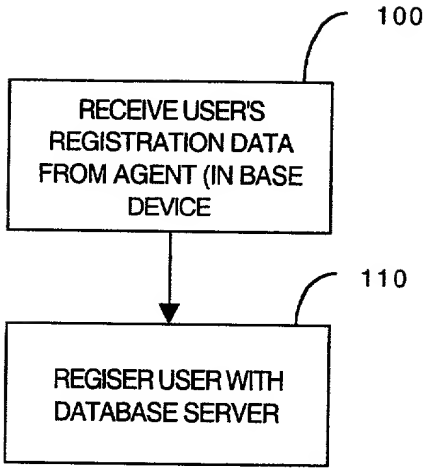
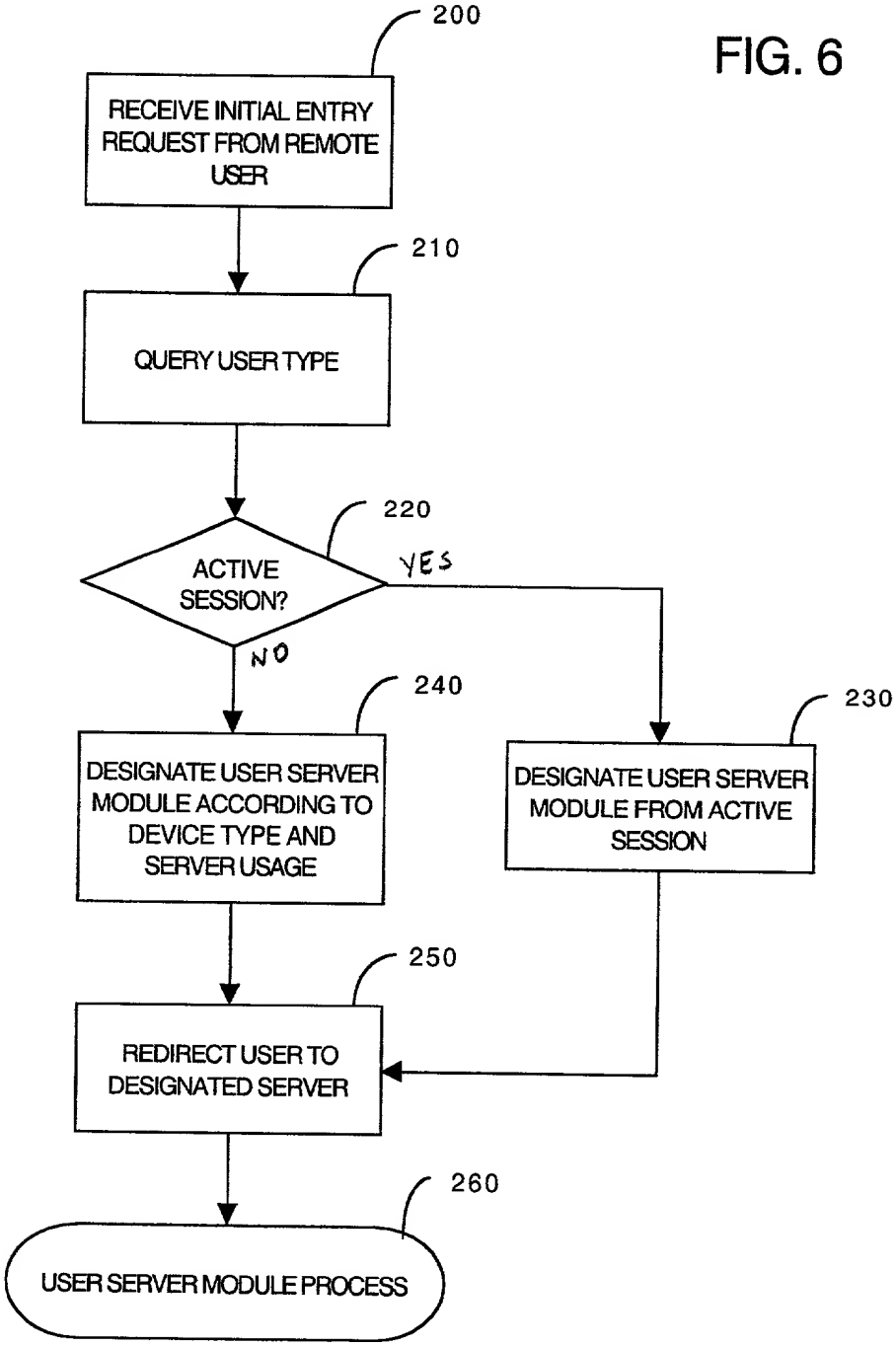

100
RECEIVE USER'S REGISTRATION DATA FROM AGENT (IN BASE DEVICE
110
REGISER USER WITH DATABASE SERVER

FIG. 6


200
RECEIVE INITIAL ENTRY REQUEST FROM REMOTE USER
210
QUERY USER TYPE
220
ACTIVE SESSION?
YES
NO
240
DESIGNATE USER SERVER MODULE ACCORDING TO DEVICE TYPE AND SERVER USAGE
230
DESIGNATE USER SERVER MODULE FROM ACTIVE SESSION
250
REDIRECT USER TO DESIGNATED SERVER
260
USER SERVER MODULE PROCESS

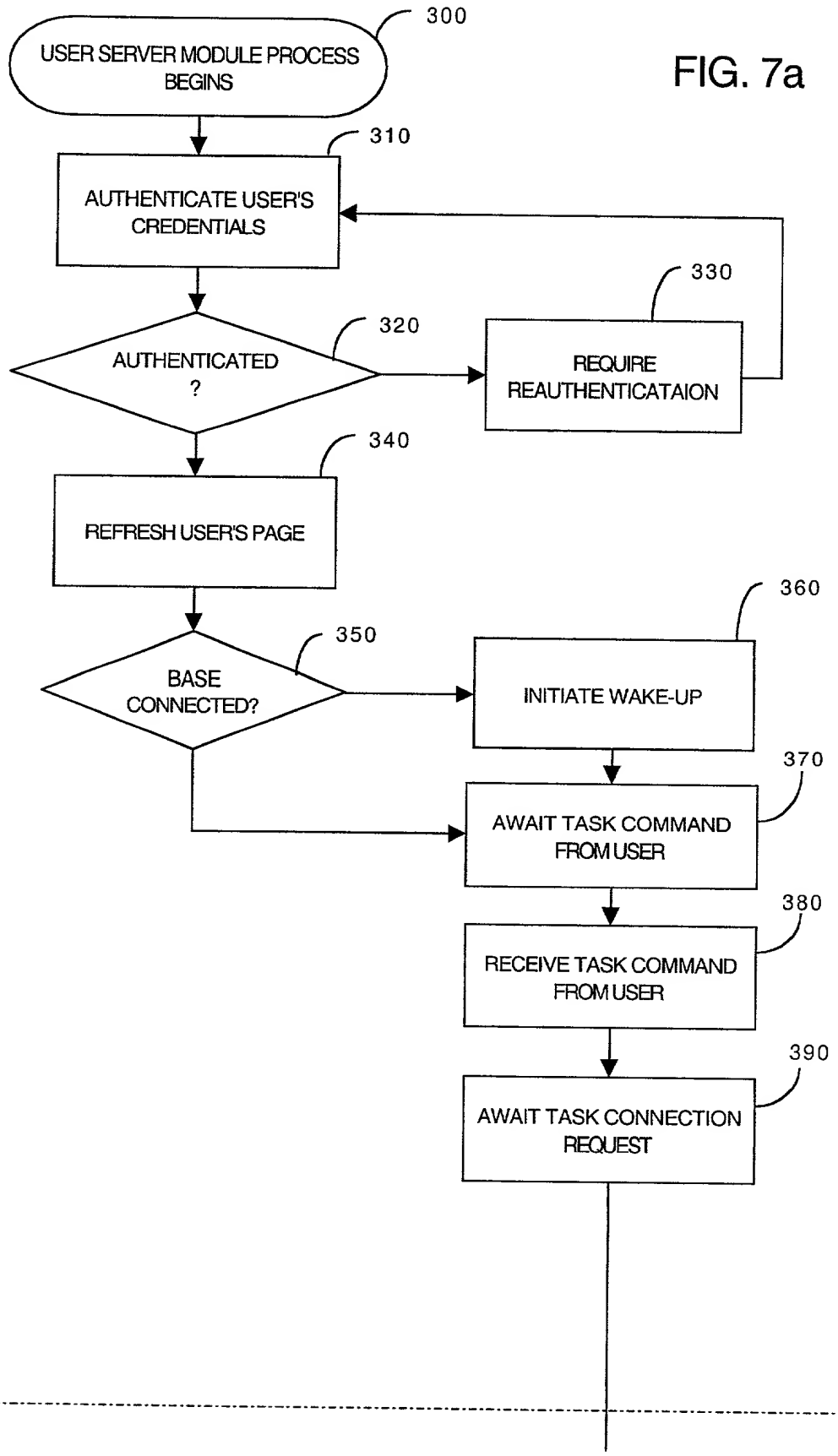

300
USER SERVER MODULE PROCESS BEGINS
310
AUTHENTICATE USER'S CREDENTIALS
320
AUTHENTICATED ?
330
REQUIRE REAUTHENTICATAION
340
REFRESH USER'S PAGE
350
BASE CONNECTED?
360
INITIATE WAKE-UP
370
AWAIT TASK COMMAND FROM USER
380
RECEIVE TASK COMMAND FROM USER
390
AWAIT TASK CONNECTION REQUEST

FIG. 7a

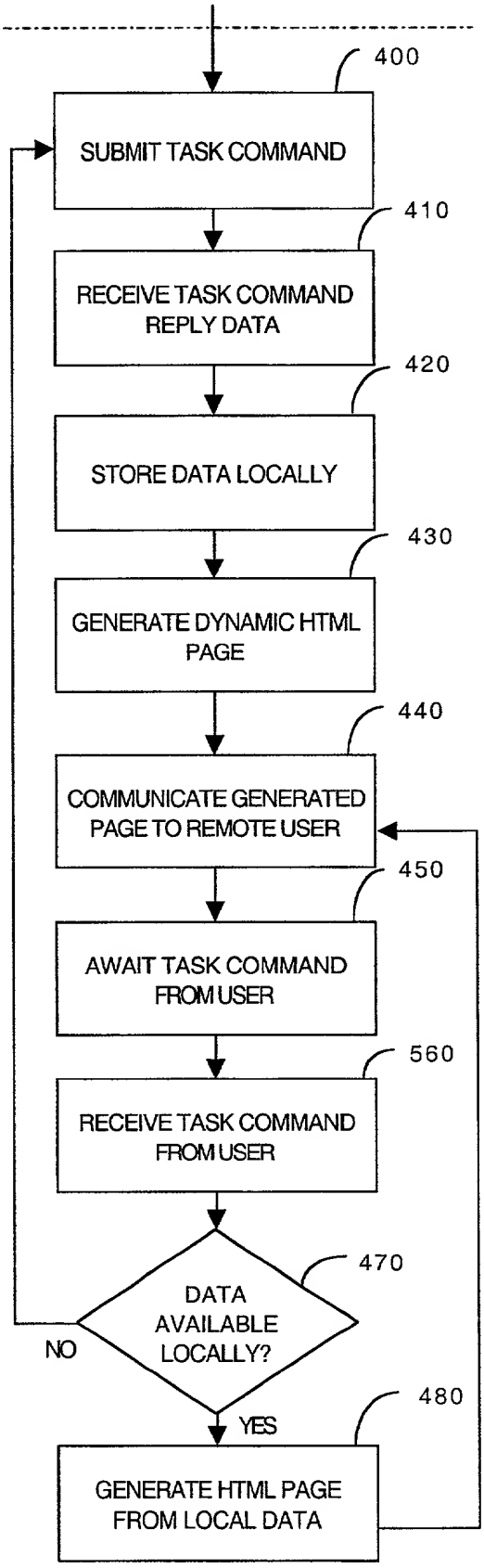

400
SUBMIT TASK COMMAND
410
RECEIVE TASK COMMAND REPLY DATA
420
STORE DATA LOCALLY
430
GENERATE DYNAMIC HTML PAGE
440
COMMUNICATE GENERATED PAGE TO REMOTE USER
450
AWAIT TASK COMMAND FROM USER
560
RECEIVE TASK COMMAND FROM USER
470
DATA AVAILABLE LOCALLY?
NO
YES
480
GENERATE HTML PAGE FROM LOCAL DATA

FIG. 7b

FIG. 8

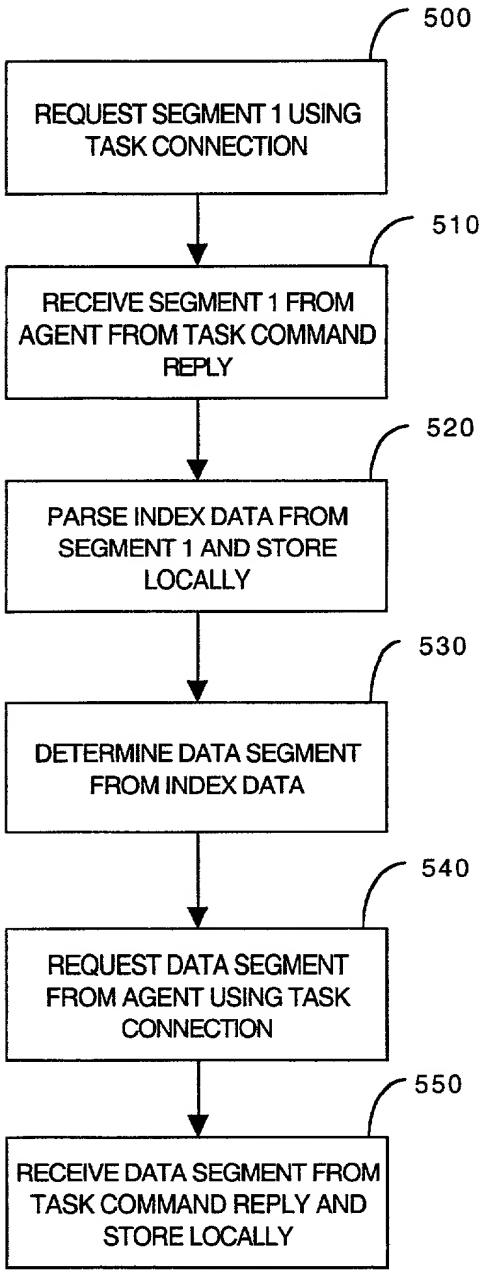
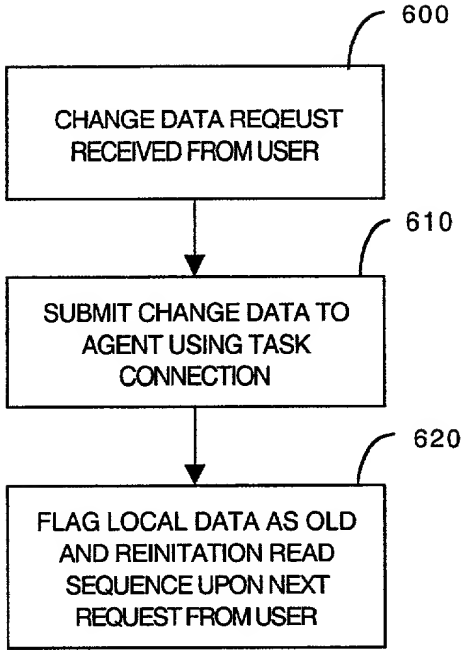

500
REQUEST SEGMENT 1 USING TASK CONNECTION
510
RECEIVE SEGMENT 1 FROM AGENT FROM TASK COMMAND REPLY
520
PARSE INDEX DATA FROM SEGMENT 1 AND STORE LOCALLY
530
DETERMINE DATA SEGMENT FROM INDEX DATA
540
REQUEST DATA SEGMENT FROM AGENT USING TASK CONNECTION
550
RECEIVE DATA SEGMENT FROM TASK COMMAND REPLY AND STORE LOCALLY

FIG. 9


600
CHANGE DATA REQEUST RECEIVED FROM USER
610
SUBMIT CHANGE DATA TO AGENT USING TASK CONNECTION
620
FLAG LOCAL DATA AS OLD AND REINITATION READ SEQUENCE UPON NEXT REQUEST FROM USER